QFD Application to a **Software-Intensive** System Development Project

Tuyet-Lan Tran
(818) 354-6174
Jet Propulsion Laboratory/Caltech
4800 Oak Grove Dr.
Pasadena, CA 91109

This paper describes the use of Quality Function Deployment (QFD), adapted to requirements engineering for a software intensive system development project, and synthesizes the lessons learned from the application of QFD to the Network Control System (NCS) pre-project of the Deep Space Network (DSN).

The selection of QFD started by recognizing that "quality" means customer satisfaction with the product. The application of QFD to the front-end of the lifecycle was built upon the premise that it is an effective technique for incorporating customer requirements into product design, in such a way that a history of improved product-releases could be achieved and quantifiable relationships could be established between customer satisfaction and the high-level specifications of a communications-and-control system.

Against this background, the adaptation of the QFD technique focused mainly on exploring the measurements of software behavior (or, performance characteristics) and on assuring the degree of completeness (or appropriateness) of high-level specifications, while facilitating the analysis and design activities of the product-development team. Where needed, QFD was coupled with other techniques, such as Yourdon's methodology.

Introduction

Developing an information-system product has never been an easy task, whether from the ground up or from a legacy system. From the product-planning perspective, such system-development activities range from understanding and capturing customer needs at the highest level to understanding product requirements and controlling at the lowest level those system-performance parameters that directly contribute to customer requirements. This paper describes the application of the Quality Function Deployment (QFD) technique to such a product-planning environment (or, pre-project phase) for the Network Control System (NCS) of the Deep Space Network (DSN).

At the time of the NCS pre-project, NASA was planning to evaluate opportunities for agency-wide consolidation and expanded commercialization of space-mission operations. In support of this new NASA horizon, the DSN at the Jet Propulsion Laboratory (JPL) started to evaluate new concepts for more autonomous space operations.

The Business Case

in the current approach to space operations and mission control, an around-the-clock operations team is needed for each DSN communications link to carry out on the ground a variety of functions, such as DSN-wide resource scheduling, spacecraft-commanding, spacecraft tracking and navigation, monitoring the state-of-health of the spacecraft and telecommunications link during a pass, ensuring correct functioning of all spacecraft and ground equipment, and coordinating failure recovery actions or "workarounds." This human-resource-intensive approach is becoming less and less viable in the future due to (1) a desire to optimize the utilization of DSN aperture, (2) a desire to accommodate Principal-

Investigator (PI)-controlled missions, (3) a desire to accommodate deep space missions with significant round-trip lig,ht-time communications delays, and (4) a desire to limit the operat i ins-t cam and DSN overall operational costs.

in the new concept of DSN operations (to be actualized via the NCS), the stated programmatic goals are to enable a net annual savings of several minim dollars in operations costs and to recover the implementation costs over a four-year payback period. The scope of the pre-project phase involves the establishment of customer needs (or requirei nents) and the concept ual design of a new network-contro l s ystem that can be rapidl y dc.vc.lope.d and deployed withi n one year of concept ual -design-complet cd.

The Selection of QFD

After some doses of Total Quality Management (TQM) rehabilitation, several of us at JPL by now believe that the most important step in product planning is to figure out who our customer(s) is/we. Unfortunatel y, this is usuall y harder than it seems at first glance. After a few weeks of system-analysis activities that did not yield any agreeable set of customers or customer requirements, it became compelling that requirements definition is a very risky business - especially when everyone wants to see a "smart system" that controls everything yet is easy to operate but cannot totally displace the worker.

The selection of QFD started by recognizing that "quality" means customer satisfaction with the product. Additionally, the QFD approach stands out as one that systematizes TQM, when c ompared to other techniques or methods for requirements definition. QFD introduces quality, indeed, as early as in the requirements phase (or "upstream") and throughout every product development stage. More importantly, '{quality" is presented from the customer's perspective, in the customer's "language" -- as opposed to the designer's or engineering perspect i ve. The goal of QFD is to deploy this customer's perspective (also referred to as Voice Of the Customer, or VOC, or customer requirements) throughout the product's technical requirements and specifications in an explicit and systematic manner. The mapping of design features or essential product-features to those customer requirements becomes a pivotal factor for QFD's role in delivering high-quality products.

For the NCS pre-project phase, a small, cross-funct ional team was appointed to carry out the QFD process. Specifically, the team's target deli verables are the product-p lanning matrices, as these represent the most important tenet for estimating, the real needs of various customers and capturing their respective requiren rots, as well as for defining the key requ i rements oft he to-be-developed product. For more det ail on QFD, refer t o [1]. A pictorial summary of the QFD's house of quality is also provided in Figure 1.

once a new technique is selected, good software management practices remind us that, in spite of its promised value, a "novel" development methodology represents a significant software risk element to a software-intensive system development project. Accordingly, a risk-mit ig at ion decision was made to adopt QFD for this pre-project phase, and to leave open the option to choose. an alternative technique for the Jwo.led, Additionally, the responsibility for "not playing by the book" (as recommended by Professor Yoji Akao himself) anti for film-tuning the technique and adapting it to the NCS was assigned to a dedicated process engineer. The process engineer selected ITI's QFD/Capture as the tool for the process.

What Made This QFD Application Unique?

In light of **NASA's** internal pressures for strategic changes, the political pressure associated with labor contracts in deployment over three different countries, and the complexities of catering to an international science community that dots not "directly buy DSN data services", the team members who were to include operations representatives were chosen for a combination of cross-functionality, knowledge and experience with the DSN, and especially respect among their peers and non-bias for their line managers or organizational affiliations. 'his combination is an absolute necessity for the team to stand a chance with the organizations that might be negatively impacted by the proposed design,

Predictably, the QFD process quickly got the team to get past the usual arguments about what customer requirements are vs. what design is. Instead, the focus shifted to:

(a) identifying and agreeing to what the various classes of customers are, and accepting that the extent of overlap in customer needs should not be used as the discriminating factor in the class partitioning activity;
(b) how to level the various customer needs;
(c) what the product is vs. what the services are (when the product is used in the "background" as a means to deliver services);
(d) how much rigor should be applied to the estimation of customer needs;
(c) how to ensure that the QFD matrices are "adequate" at capturing, the "essential product features" that correlate with the customer needs/requirements;
(f) how to ensure the completeness of a minimalist set of product requirements -- whether with or without QFD.

Combining the Yourdon methodology with the QFD process was critical to the successful completion of the NCS cone.cptuai design. The combined process included 5 distinct activities, which are embedded along a spiral model itself. Of these 5 activities only the development of operations scenarios is not included in the following discussion.

1. Defining the. Context

This activity is similar to the development of the Yourdon's context diagram. The objective is to define the solution space, in terms of a black box that interacts with its environment. The black box is intended to represent the product or scl~'ices-t[)-l)c-- c(~[lsll{l~c.{l by the end-user.

This diagram forms the starting point for developing the conceptual design of the NCS. See Figure II for NCS's context diagram.

2. Developing Level-1 QFD Matrix

At this pre-project phase, QFD identifies the key aspect of NCS that the customer is interested in from the customer's perspective (in this case, the Decision-makers) by filling-in the columns headed "Whats" of Figure I. This consists of eliciting expected customer needs (via interviews) and estimating real customer needs, via deployment scenarios that are derived from the context diagram and that assume expansions of selected external-interfaces. Besides having the context diagram as the input, this activity is very QFD-specific. for more detail on the application of the QFD technique, refer to [1] and [2].

Where the NCS's application is unique is in the coupling of the level-1 DFD with the QFD matrix, for the filling-in of the columns headed "How" of Figure 1 (or product requirements). 'The level-1 DFD is used, indeed, for the:

- identification of essential software processes that make up the product which will solve the customer needs (or the "whats");
- specification of the performance parameters associated with the primary data flows of these processes;
- capturing of decision-tree parameters.

The NCS level-1 QFD is provided in Figure III. This QFD's correlation matrix makes technical risk identification explicit, by highlighting potential performance bottlenecks and setting the basis for subsequent risk assessment.

The power of QFD is founded in how it asks the team to consider alternative solutions, to make tradeoff decisions explicitly, and how it efficiently captures decisions and analysis of information among QFD team members, as well as with other project-team members for continuous in-project or across-project refinements and improvements.

3. Developing Level-1 Data Flow Diagram

This activity started after the start of the level-1 data flow diagram (DFD), that identifies two software end items; namely, the Customer Service and Scheduling (CSS) and the Equipment Activity Controller (EAC). It was agreed that both CSS and EAC needed their respective level-1 DFDs. For the purpose of this paper, only the CSS's DFD is provided in Figure IV.

The objective of the level-1 DFD is to develop a logical model of the CSS, in terms of processes, control flows, and data flows (or messages). This model should be high-level enough to allow for consistency check and validation against user requirements and the level-1 QFD; and, to necessitate a next level of decomposition (into a level-2 DFD). Figure IV shows the software architecture of the CSS, and has evolved from an earlier version of software-requirements DFD. The earlier DFD facilitated analysis, and was critical to the verification of completeness of processes. The latter version (as captured in Figure IV) sets the foundation for further, detailed design or for prototyping of the CSS.

4. Rapid Iteractions for Cost Estimation

The combination of the context diagram, level-1 QFD, level-1 DFD, operations scenario and associated hardware diagrams (not covered in this paper) were adequately detailed, to support the development of a product-based Work Breakdown Structure (WBS), a level-1 schedule, and a full cost-estimation effort.

It is worth noting that the tool environment established and controlled by the process engineer was a key factor in the successful definition and application of a rigorous process. Rigor-enforced, in turn, enabled the rapid iterations and refinements of the NCS customer and product requirements without generating reams of cumbersome and distracting documentation. Finally, the accelerated rate at which a cross-functional, co-located QFD team could iterate and release the level-1 QFD became an effective catalyst for the problems of requirements volatility and accurate problem-domain knowledge acquisition noted by Curtis et al [3].

Conclusion

This paper has described how the QFD process has been adapted and mixed-in-with Yourdon's methodology, to enable the capture of customer requirements and rapid definition of a software-intensive system's conceptual design. The author believes that progress toward the goal of having a methodology that enforces rigor in requirements capture and conceptual design, as will as in their systematic reuse, lies in a development support environment, which includes QFD-based tools (for the explicit and disciplined incorporation of customer requirements into product design). This would necessitate further bridging, between QFD-based tools and other operations-scenario development and simulation tools. The contribution of the QFD-based technique is in the front-end of the process, and has most impact in (a) enabling optimal decision making by the right QFD development team; (b) enabling rapid customer-centered model development, checkout, modification, and reuse. Moreover, results to-date suggest that QFD could be a valuable tool for the entire life-cycle, for controlling requirements changes, iterating descope-alternatives, and capturing the results of those decisions in a way that would positively impact subsequent phases of development in significant ways.

Acknowledgments

The results reported in this paper were carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. Special thanks are given to the entire NCS QFD team; and, in particular to John Leflang, manager of the NCS Pre-project, for having accepted the author's recommendation for using QFD, to Thang Trinh and Monica Wang for their insightful inputs on what "hiccups" they are constantly having, when "applying" QFD using the QFD/Capture tool, and to Cesar Labarca and Joseph Sherif for their editorial assistance with this paper.

References

[3] Curtis, B., Krasner, H., and N. Iscoe, "A Field Study of *The Software Design Process for Large Systems*," Comm. ACM, 31,11, pp. 1268 - 1287, 1988.

[2] Tran, T.L.. and J.S. Sherif, *"QFD: An Effective Technique for Requirements Acquisition and Reuse,"* Proceeding of Second IEEE International Software Engineering Standards Symposium, PP. 191 - 200, August 1995

[1] Yoshizawa, T., Togari, H., and T. Koribayashi, *"QFD, Integrating Customer Requirements into Product Design,"* (Akao, Y. editor), Productivity Press, Cambridge, MA 1990.

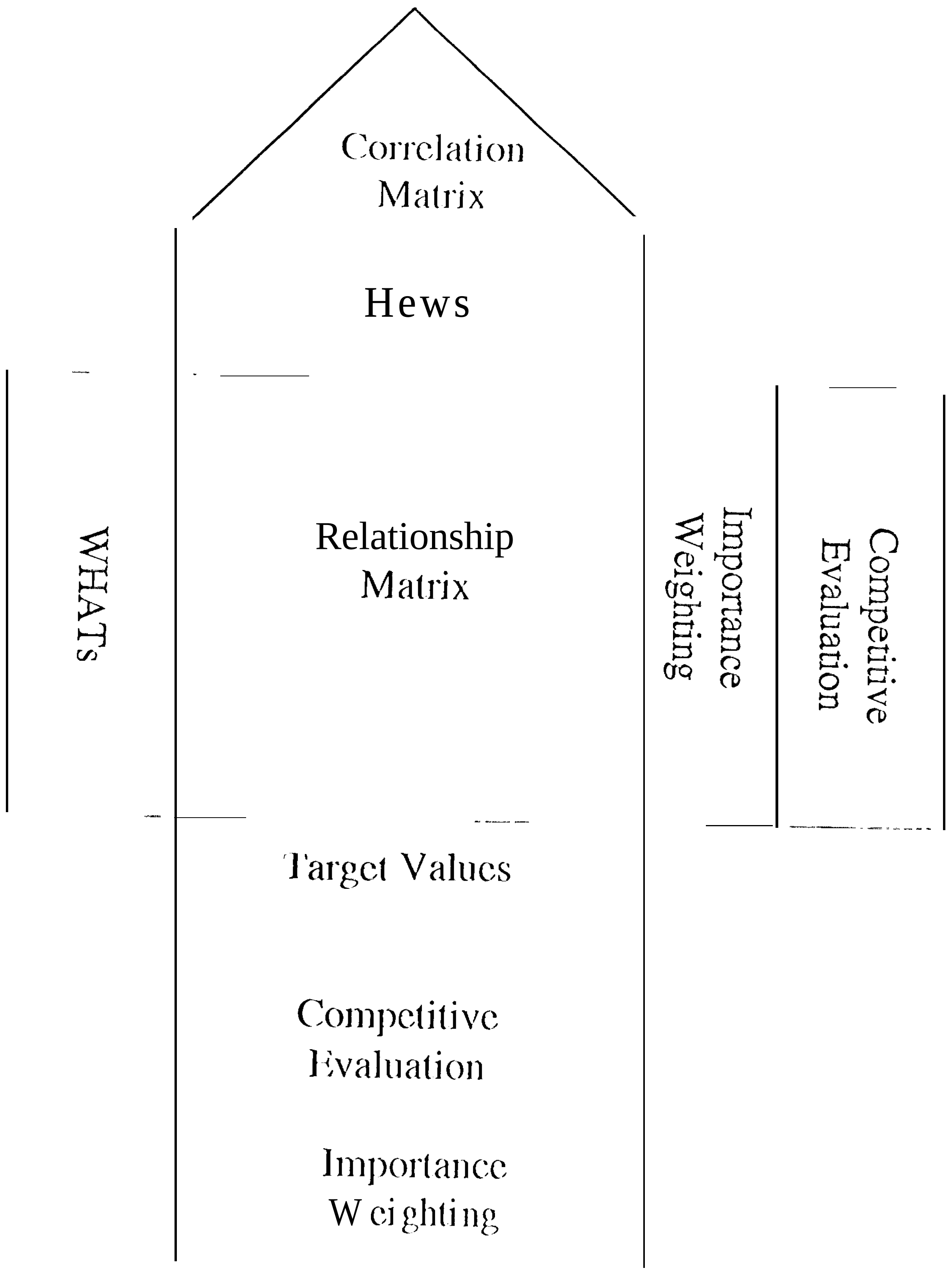
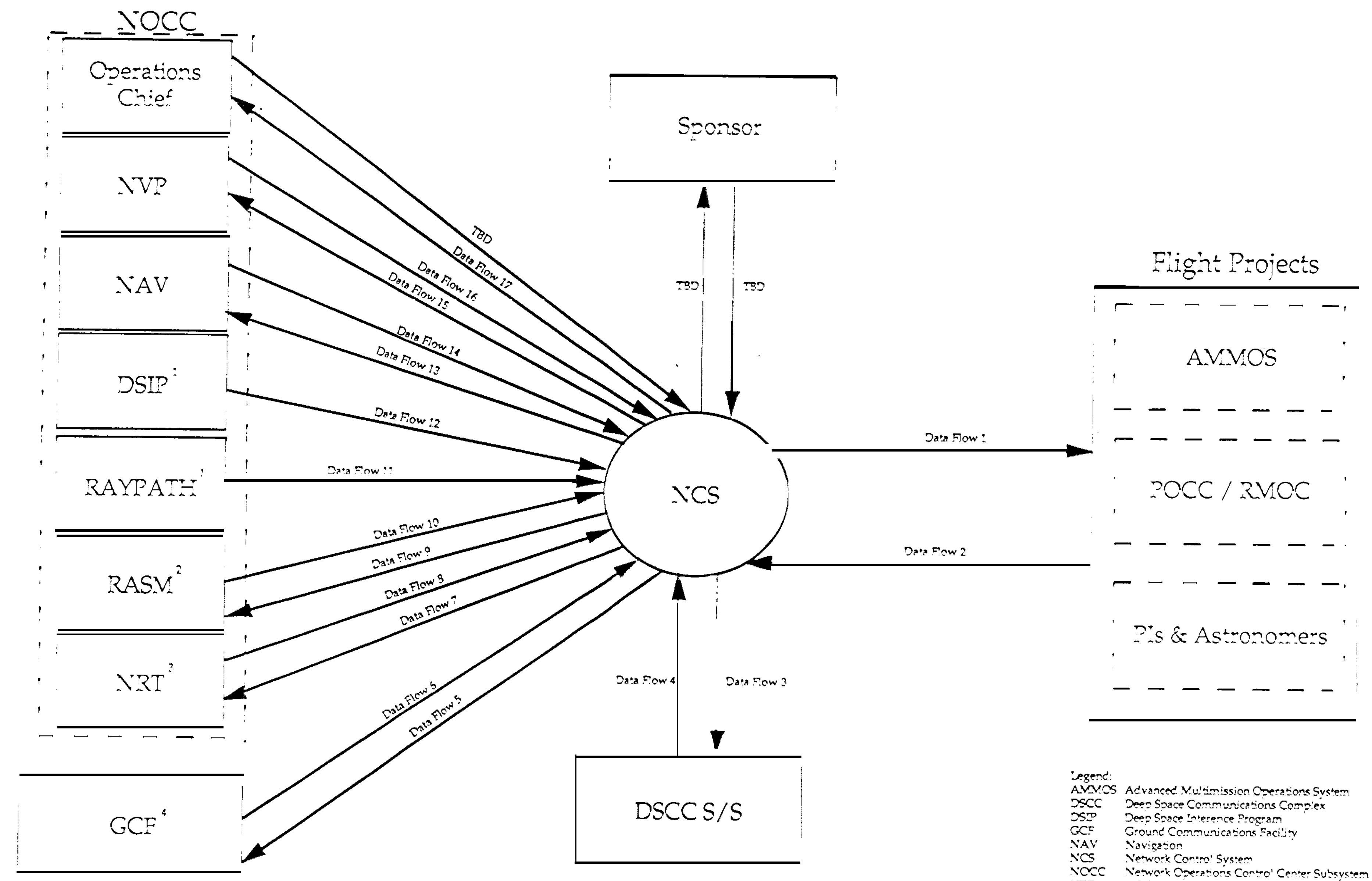


Figure 1. Overview of QFD Chart or Matrix



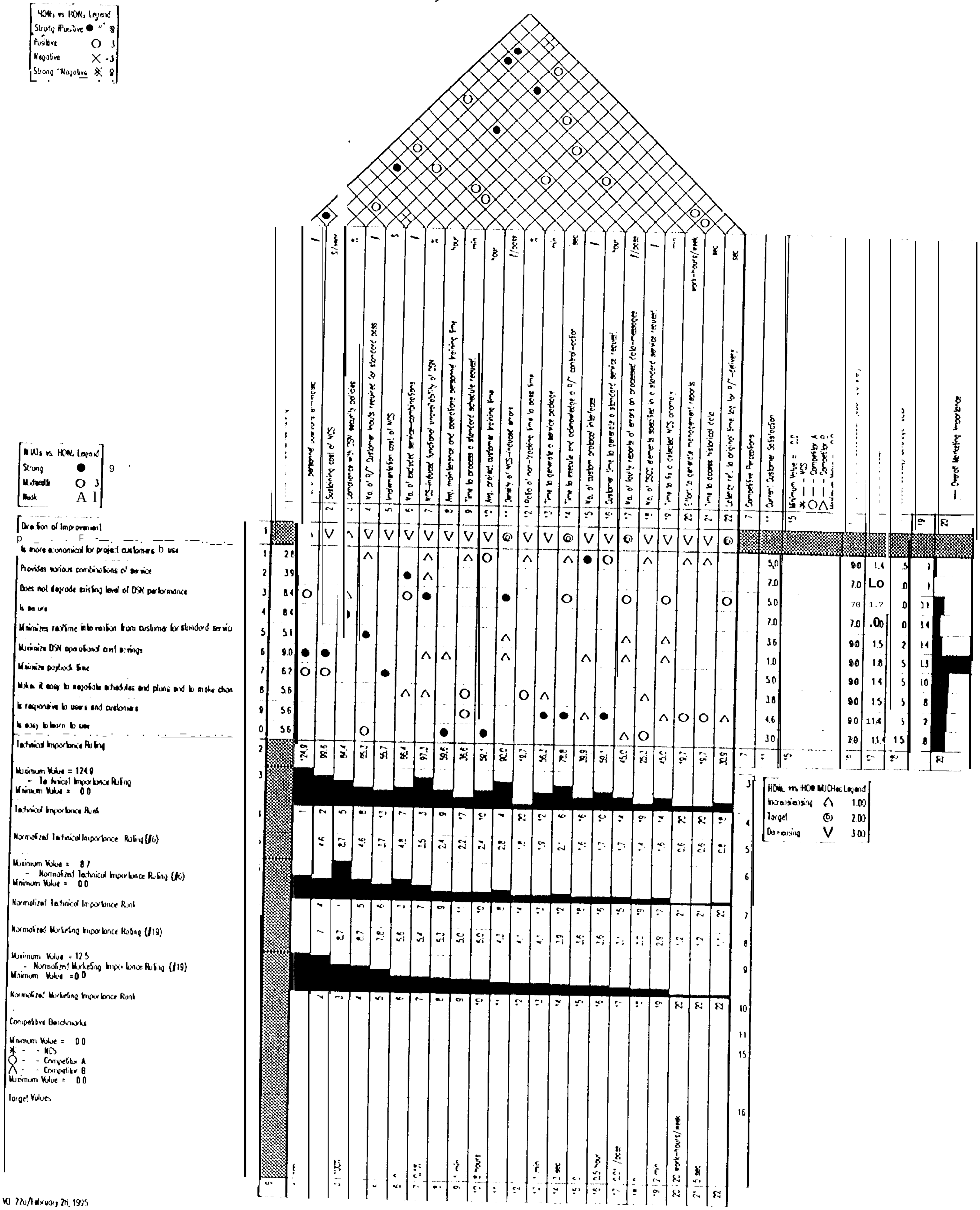
Legend:

AMMOS	Advanced Multimission Operations System
DSCC	Deep Space Communications Complex
DSIP	Deep Space Interence Program
GCF	Ground Communications Facility
NAV	Navigation
NCS	Network Control System
NOCC	Network Operations Control Center Subsystem
NRT	NOCC Real Time Subsystem
NVP	NOCC VLBI Processor
PIs	Principal Investigators
POCC	Project Operations Control Center
RASM	Remote Access Scheduling Mailbox
RMOC	Remote Mission Operations Center

Notes:
1. The functions of DSIP and RAYPATH will be integrated with NCS.
2. Interface to RASM will be deleted.
3. All existing interface to NRT except NEL (used for communication protocol) will be deleted.
4. Demonstrated here for communication protocol use only.
5. Data Flows 1 to 17 - refer to NCS Interface Table for details.

Figure II. NCS Context Diagram

Figure III - NCS LEVEL - 1 QFD



HOWs vs HOWs Legend
Strong Positive ● 9
Positive O 3
Negative X -3
Strong Negative ⊗ -9
WHATs vs HOWs Legend
Strong ● 9
Moderate O 3
Weak Δ 1
HOWs vs HOW MUCHes Legend
Increasing Δ 1.00
Target ◎ 2.00
Decreasing V 3.00
Direction of Improvement
1 Personnel positions eliminated
2 Sustaining cost of NCS $/year
3 Compliance with DSN security policies
4 No. of R/T Customer inputs required for standard pass
5 Implementation cost of NCS $
6 No. of excluded service-combinations
7 NCS-induced functional unavailability of DSN
8 Avg. maintenance and operations personnel training time hour
9 Time to process a standard schedule request min
10 Avg. project customer training time hour
11 Density of NCS-induced errors #/pass
12 Ratio of non-tracking time to pass time
13 Time to generate a service package min
14 Time to execute and acknowledge a R/T control-action sec
15 No. of custom protocol interfaces
16 Customer time to generate a standard service request hour
17 No. of faulty reports of errors on processed data-messages #/pass
18 No. of DSCC elements specified in a standard service request
19 Time to fix a detected NCS anomaly min
20 Effort to generate management reports work-hours/week
21 Time to access historical data sec
22 Latency rel. to original time tag for R/T-delivery sec
Competitive Perceptions
Current Customer Satisfaction
Minimum Value = 0.0
Overall Marketing Importance
Is more economical for project customers to use 2.8
Provides various combinations of service 3.9
Does not degrade existing level of DSN performance 8.4
Is secure 8.4
Minimizes realtime interaction from customer for standard service 5.1
Minimize DSN operational cost savings 9.0
Minimize payback time 6.2
Makes it easy to negotiate schedules and plans and to make changes 5.6
Is responsive to users and customers 5.6
Is easy to learn to use 5.6
Technical Importance Rating
Maximum Value = 124.9
Technical Importance Rating
Minimum Value = 0.0
Technical Importance Rank
Normalized Technical Importance Rating (#6)
Maximum Value = 8.7
Normalized Technical Importance Rating (#6)
Minimum Value = 0.0
Normalized Technical Importance Rank
Normalized Marketing Importance Rating (#19)
Maximum Value = 12.5
Normalized Marketing Importance Rating (#19)
Minimum Value = 0.0
Normalized Marketing Importance Rank
Competitive Benchmarks
Minimum Value = 0.0
* - NCS
O - Competitor A
Δ - Competitor B
Maximum Value = 0.0
Target Values
1 min
8 hours
1 min
2 sec
0.5 hour
0.01 /pass
2 min
2 work-hours/week
5 sec

	Importance	Current Customer Satisfaction				
Is more economical for project customers to use	2.8	5.0	9.0	1.4	.5	[illegible]
Provides various combinations of service	3.9	7.0	7.0	Lo	.0	[illegible]
Does not degrade existing level of DSN performance	8.4	5.0	7.0	1.?	.0	11
Is secure	8.4	7.0	7.0	.0o	0	14
Minimizes realtime interaction from customer for standard service	5.1	3.6	9.0	1.5	2	14
Minimize DSN operational cost savings	9.0	1.0	9.0	1.8	5	13
Minimize payback time	6.2	5.0	9.0	1.4	5	10
Makes it easy to negotiate schedules and plans and to make changes	5.6	3.8	9.0	1.5	5	8
Is responsive to users and customers	5.6	4.6	9.0	11.4	5	2
Is easy to learn to use	5.6	3.0	7.0	11.[illegible]	1.5	.8

HOW	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
Technical Importance Rating	124.9	99.6	84.4	65.3	55.7	66.4	97.3	59.6	36.6	56.1	90.0	19.7	56.1	78.8	39.9	59.1	45.0	25.3	45.0	19.7	19.7	30.9
Technical Importance Rank	1	2	5	8	13	7	3	9	17	10	4	20	12	6	16	10	14	19	14	20	20	18

VO 22b/February 28, 1995

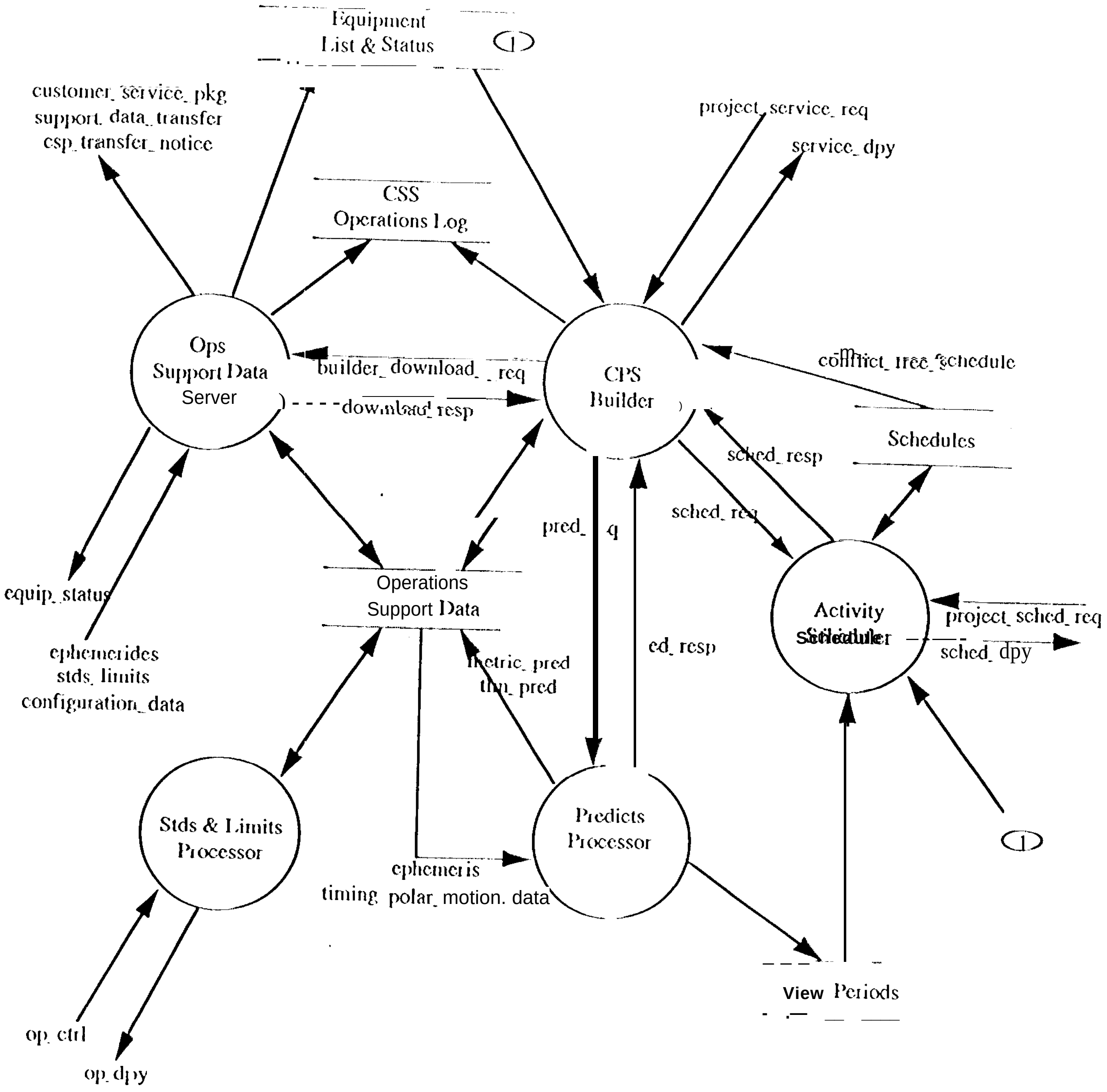


Figure IV. NCS CSS Software Architecture